L2F - Legacy to the Future Framework

Denise C. George (T-10) and James E. George (ACL)

The purpose of L2F, Legacy to the Future Framework is to provide an integrated modeling system of new and legacy code components that builds upon the strength in scientific modeling and simulation of Los Alamos National Laboratory. Existing modeling systems have been narrow in focus, and efforts to link existing codes have been tedious and uncommon. However, recent advances in computational tools enable a more general solution to this problem.

L2F is implemented using JAVA and CORBA. JAVA is a platform-neutral object-oriented language with built in networking and database connectivity. It allows sending database queries (SQL) to any database; i.e., the same program can be used to connect to ORACLE and SYBASE. CORBA

...we can incorporate legacy codes written in many different languages and that run on a variety of platforms along with new codes into our Framework.

(Common Object Request Broker Architecture) provides platform transparency and language transparency. It specifies the functionality of the ORB which is a software "object bus." CORBA lets objects transparently make requests to and receive responses from other objects located anywhere on the network. The interface between objects is defined using the language IDL. It defines both the data and the services on the data and serves up objects on the platform from which the request has been made.

By using JAVA and CORBA, we can incorporate legacy codes written in many different languages and that run on a variety of platforms along with new codes into our Framework. The Framework is designed to be platform neutral and language neutral, to provide security and confidentiality, to be universally accessible via the internet, to incorporate distributed components, and to be easily augmented via a 'plug and play' design. This design provides these capabilities and features:

- Chaining of low-level models to produce integrated models.
- Secure inter and intra-national collaborations.
- Minimal amount of work to add a new model or component to the Framework.
- Catalogue of run histories.
- High performance by running models on their native platforms.
- Model owners control over access to owner's model's data.
- Data movement only as necessary.

L2F consists of a setup server and its associated run-history data, application servers, clients and optional authentication and security services. Adding an application to the Framework requires writing its input specification using the framework grammar and implementing its application server according to the defined framework IDL. The setup server and client remain unchanged. L2F has been implemented on a set of related codes native to UNIX and PC platforms using both ORACLE and ACCESS databases

The Setup Server provides a simulation specification from which the Client creates a graphical user interface. Using this GUI the user can locate, select, and edit an entry or prepare a new entry in the Run-History Data Base, or request the Application Server to start a simulation, visualize results or select among other supported actions. The Run-History Data Base collects simulation histories including input specification and links to other input and resulting outputs. It also keeps a record of which Application Server was used in a past simulation or will be used in a future simulation run.

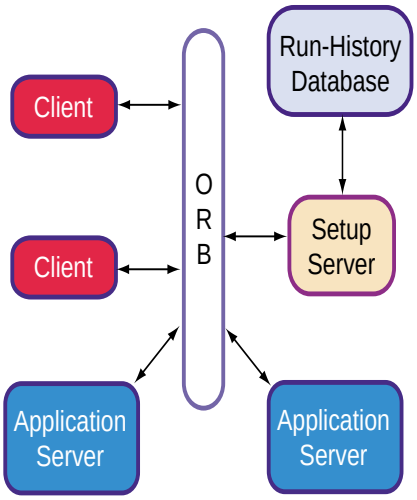

Figure 1: Schematic relationships of the components of the L2F framework.

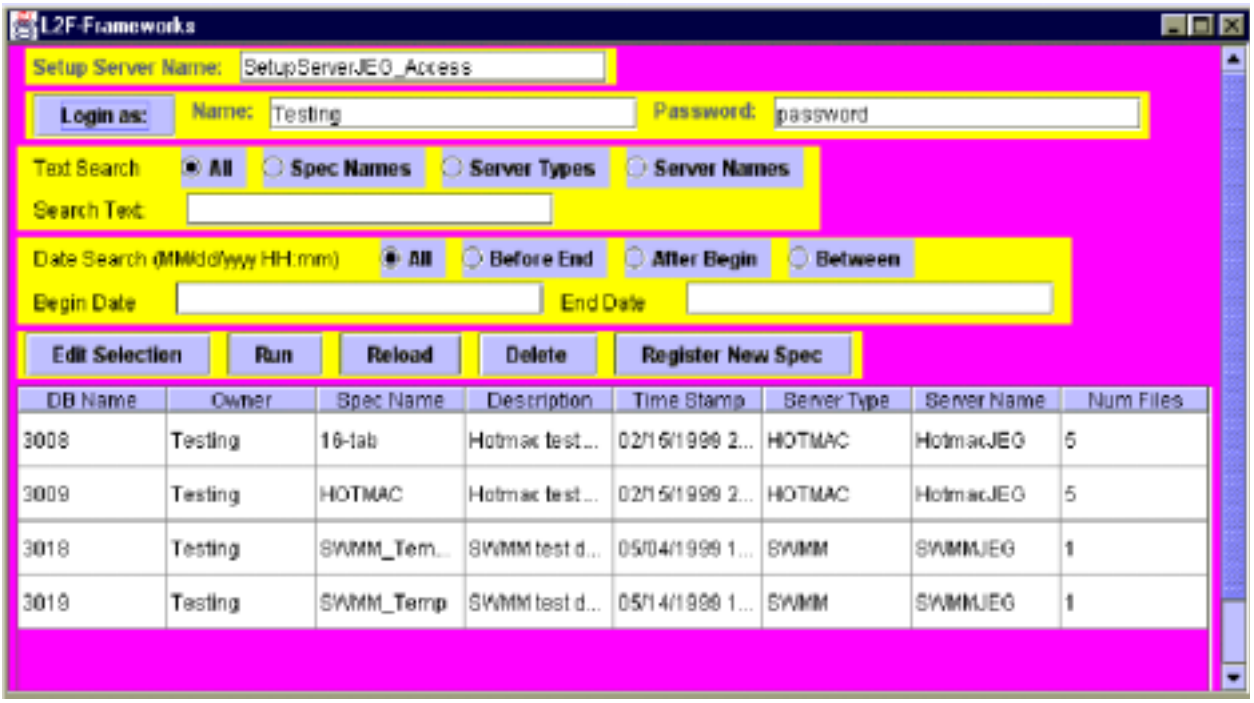

Figure 2: L2F login screen showing available simulation specifications available in the Run-History Database. Choosing a specification will allow the client to generate the GUI for that application. Subsets of specifications may be chosen by using either the text- or date-search options.

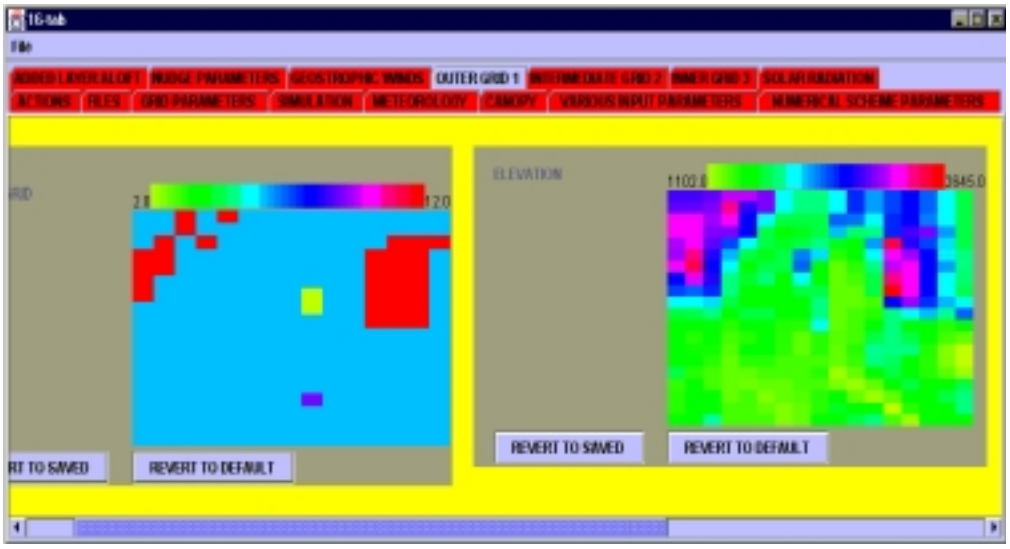

Figure 3: GUI created automatically for an atmospheric simulation showing allowable actions.

[1] J. E. George, D. C. George, "L2F - Legacy to the Future Framework," presented at ACM 1999 Java Grande Conference Workshop for High-Performance Network Computing, June 12-14,1999, San Francisco, CA., (LA-UR-99-2361, Los Alamos National Laboratory).